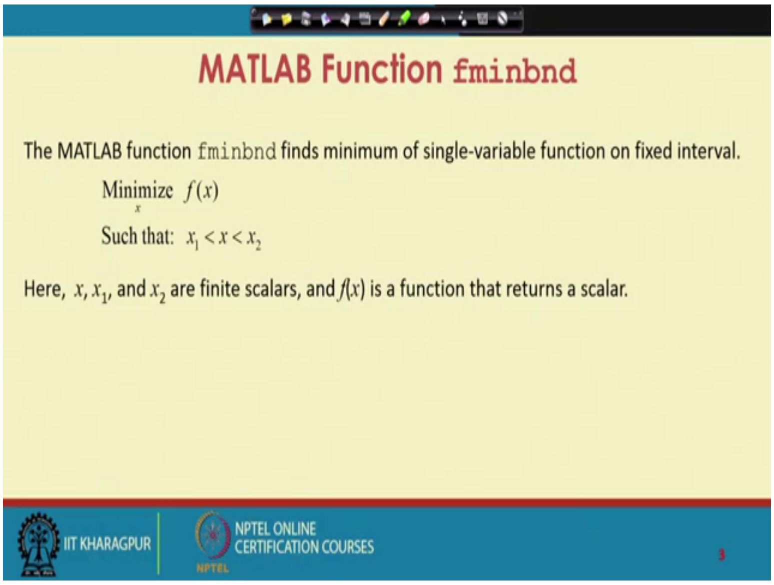
Optimization in Chemical Engineering
Prof. Debasis Sarkar
Department of Chemical Engineering
Indian Institute of Technology, Kharagpur

Lecture – 53
Applications of Optimization
(Contd.)

Welcome to lecture 53. This is week 11 and in this week, we were talking about Applications of Optimization. So, as of now, we have seen certain examples from Chemical Engineering and solved those optimization problem using MATLABs optimization toolbox. So, we have now become familiar with some of the functions that are available in MATLAB optimization toolbox.

In today's lecture, we will learn 1 more function known as fminbnd; fminbnd solves a minimization problem of a single variable function and the optimum is located in a fixed interval; that means, you know the bounds on the decision variable. So, let us see how we can solve a minimization problem in single variable using fminbnd.

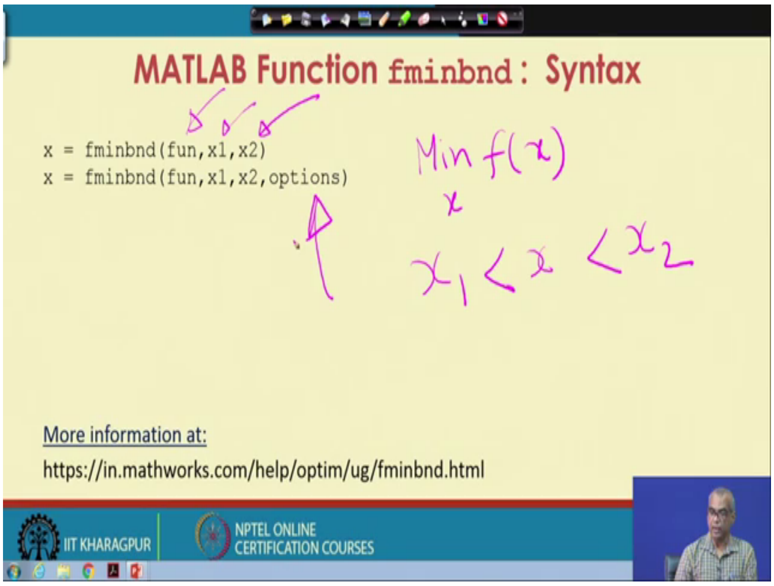
(Refer Slide Time: 01:20)



So, the MATLAB function fminbnd finds minimum of a single variable function on fixed interval here x 1 x 2 are finite scalars and f x is a function that it returns a scalar.

So, basically we have a single variable function and we are going to find the minimum of the function in the fixed interval x 1 to x 2; x, x 1, x 2 are all scalars and f x is a function that will also return a scalar value.

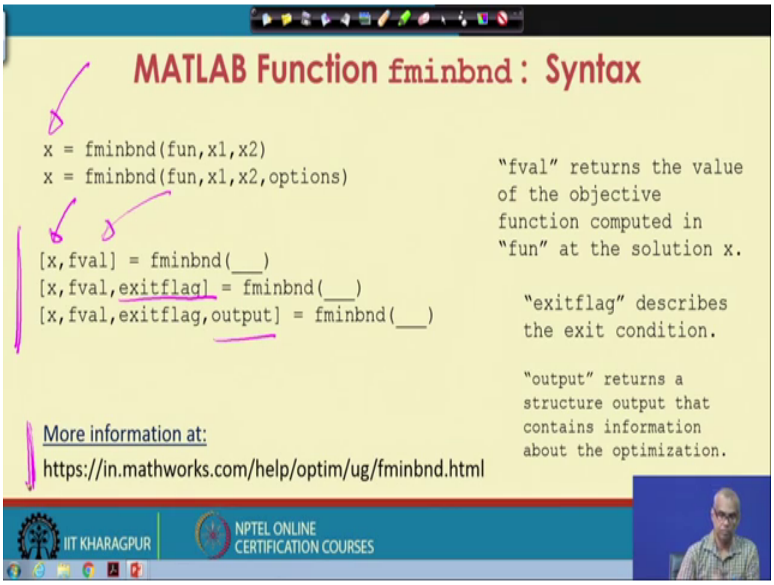
(Refer Slide Time: 02:13)



So, this is a syntax for fminbnd; x equal to fminbnd and the arguments are fun x 1, x 2. So, the fun is the name of the function, where you define the objective function and x 1, x 2 at the lower bound and upper bound on the decision variable.

So, basically it solves minimization f x and x is bounded between x 1 and x 2. You can also supply the options argument to give instruction to fminbnd for additional work such as you may like to display the values that have been obtained at each iterations.

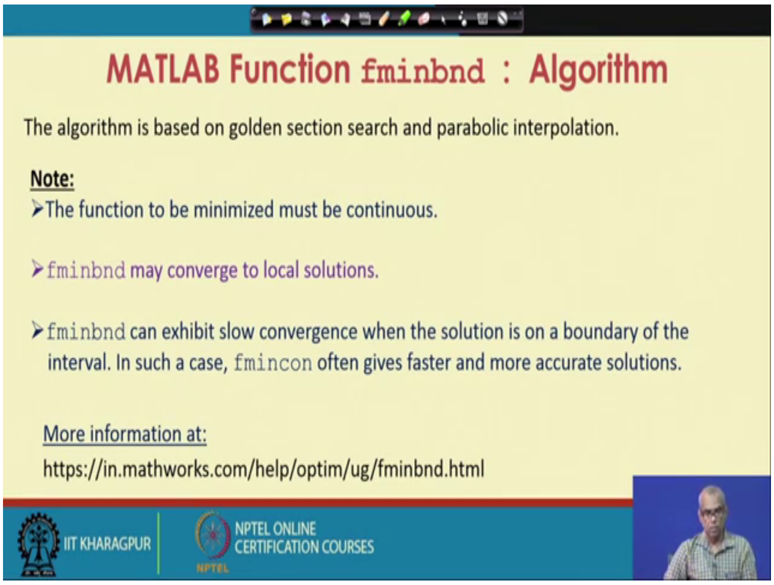
(Refer Slide Time: 03:31)



Now, so, this syntax tells you that not only you get the final solution, but you also get the function value. That means value of the objective function computed at the solution x. You can also have as an output exit flag which describes the exit conditions; how the optimization was terminated. And finally, you may also have a structure called output as output variable which will contain information about the optimization.

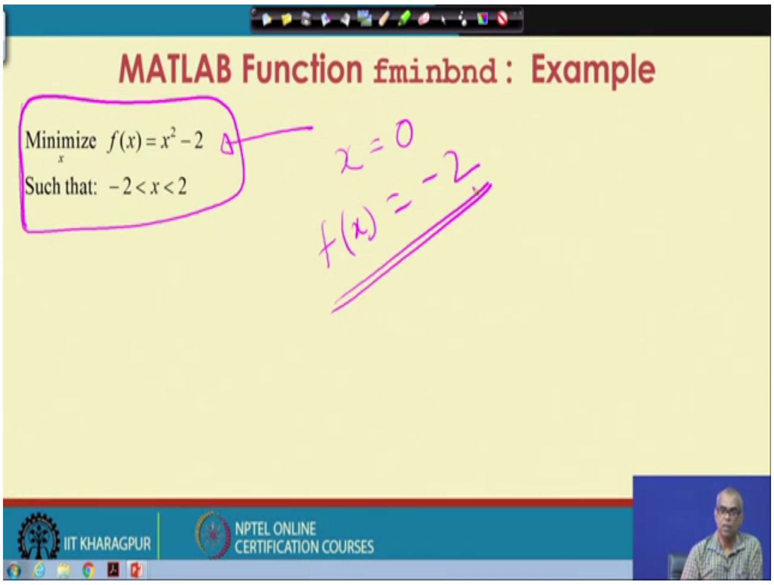
So, the output structure will tell you about the informations on the optimization; how many function values where required; what algorithm was used so on and so forth. You can obtain more information about the use of fminbnd in this MATLAB website.

(Refer Slide Time: 05:09)



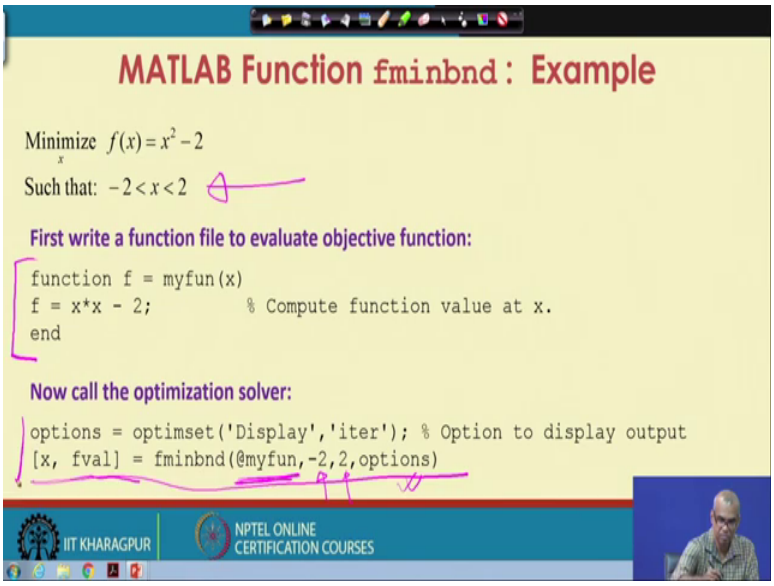
Fminbnd uses an algorithm that is based on golden section search and parabolic interpolation. So, the algorithm on which fminbnd works is based on golden section search and parabolic interpolation. Certain points to be noted about fminbnd the function to be minimized must be continuous. fminbnd make converge to local solutions. fminbnd can exhibit slow convergence, when the solution is on a boundary of the interval. In such case fmincon often gives faster and more accurate solutions.

(Refer Slide Time: 06:19)



Now, let us take a simple example to understand the use of the function fminbnd. So, you want to minimize the function f x equal to x square minus 2, and x is bounded between minus 2 and plus 2. So, this is the problem we want to solve using fminbnd. You can find out the solution by just looking at the problem. The minimum value that the function will have is when x equal to 0 in that case function value will be minus 2; note that the function value cannot go below this because x square is always a positive term. So, the minimum value will be minus 2 and that corresponds to the value of x equal to 0. So, let us see whether we get this solution using fminbnd.

(Refer Slide Time: 08:09)



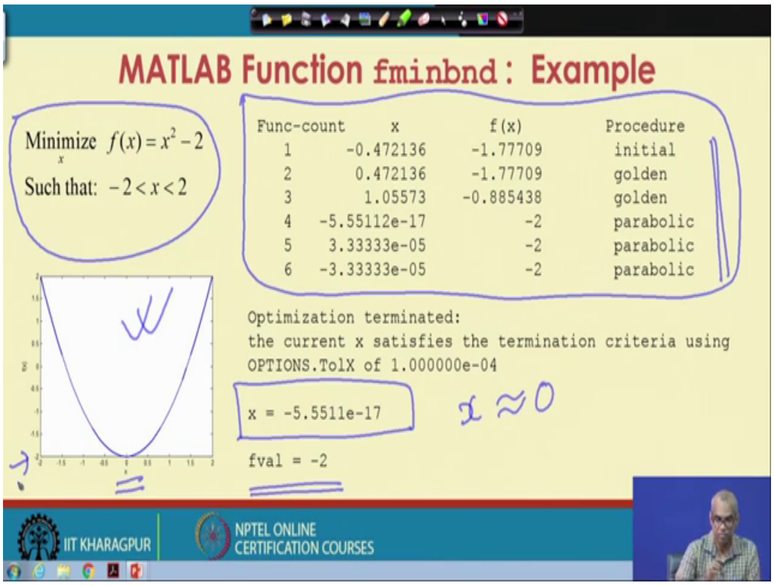
So, as usual we have to first write a function file to evaluate the objective function. So, that is the argument f u n. So, we need first to write a function file, where you will define the objective function. So, you have to follow the rule of writing a function and this is how you write the function file function f equal to my fun argument x.

So, name of the function is my fun. So, within this function I define the objective function; f equal to x square minus 2. So, this function receives x as input and sends f which is the objective function value as output. Now, we are ready to call the f min bound fminbnd optimization solver for minimization. So, we can define options equal to optimset 'Display', 'iter'.

So, it will display the output at each iteration and then use the syntax to call fminbnd. So, the output will be x and fval that means, the solution as well as the function value at this solution. Note how this name of the function is being applied minus 2 is the lower bound plus 2 is the upper bound on the decision variables and the options.

So, this is how you will call the optimization solver. You can write this 2 lines on MATLAB command window. Before that of course, you will create a function file and save this and then, either you can write these 2 lines on the MATLABs command window or you can write another script file and write this thing save this and then run.

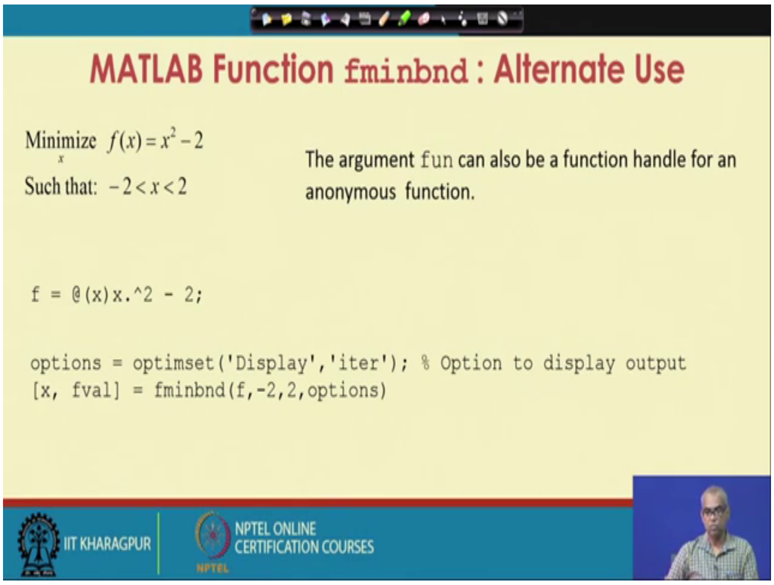
(Refer Slide Time: 11:20)



So, this is what you get as solution. Since we have opted to display the output at each iterations, the MATLAB sends you the output at each iteration. And these are the algorithms that has been used at various iterations. It started with golden section search and then, later on move to parabolic interpellation. So, this is actually a very efficient way of doing one dimensional search; you can start with a golden section and then move to cubic interpolation or parabolic interpolation; fminbnd uses golden section search and parabolic interpolation. So, the solution that we get is minus 5 into 10 to the power minus 7.

So, this is almost 0 and the function value as expected is minus 2; is a single variable function. So, we can simply plot this function, as we have done here and you can locate the minimum at x equal to 0 corresponds to f x equal to minus 2. So, obviously, you get the same solution.

(Refer Slide Time: 13:36)

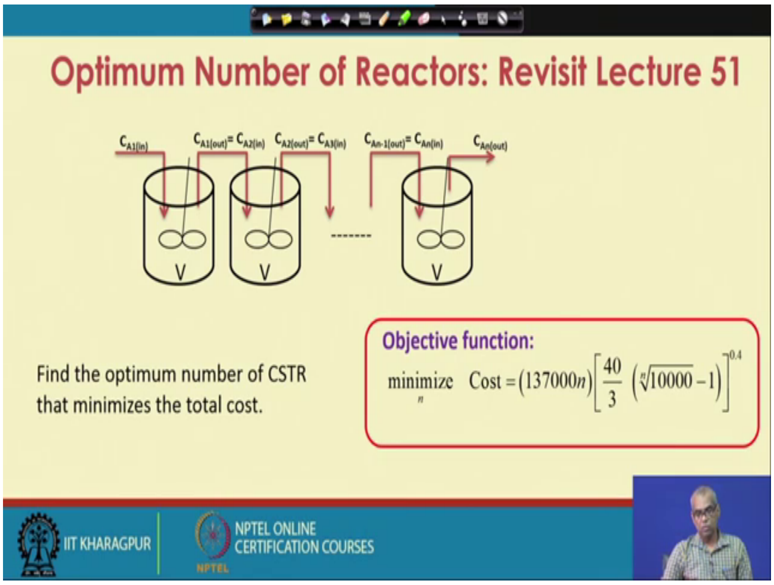


Now, there is an alternate use look at the previous way of calling the solver and particularly look at this argument at my fun. So, this is the function which contains the objective function value.

So, my fun computes the value of the objective function at decision variable value x and supplies it to fminbnd solver. Now, alternately what we can do is the argument fun can also be a function handle for an anonymous function. For a simple function like this f x equal to x square minus 2, I can define f as soon f equal to add x; then x square minus 2 and you can now write this f here. You can also write this here. So, this is sometimes convenient, if you have a simple function.

Obviously, if you have a complicated functions, you will define a function file like my fun and save it as my fun dot m and then, follow the syntax that we have seen in the previous slides.

(Refer Slide Time: 15:55)

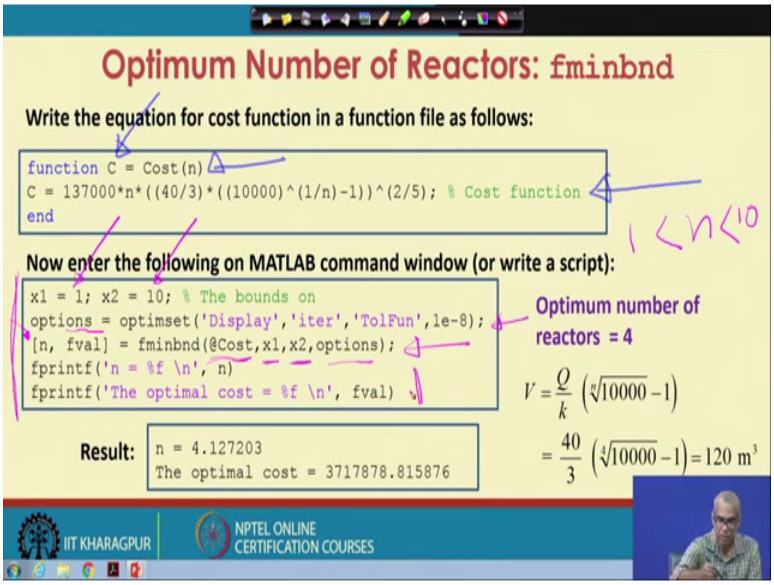


So, now let us try to solve some of the problems that we have already solved in lecture 51 using solvers such as f min search and f min con. So, the first problem that you take is optimum number of reactors in a reactor train. So, you have n number of CSTRs in series in a reactor train. We want to find the optimum number of CSTR that minimizes the total cost.

So, you can go back to the slides of lecture 51 and can see the optimization problem formulation. Please recall that you are able to express the total cost as a function of n; where n is number of CSTR in the series. So, as n changes the cost changes and given the data and information, we are able to express the cost as a function of n.

So, the total cost is now function of single variable n, which is the number of CSTR units. So, we should be able to solve it using fminbnd. Of course, you have to put a lower bound and upper bound.

(Refer Slide Time: 18:01)



So, let us look at the MATLAB code that you have to use. So, first write the function file for the objective function. So, we will define a function known as Cost. You can define by any other name you like. So, we define a function called Cost which takes n as input; n is nothing but my decision variable number of CSTR units in the reactor train.

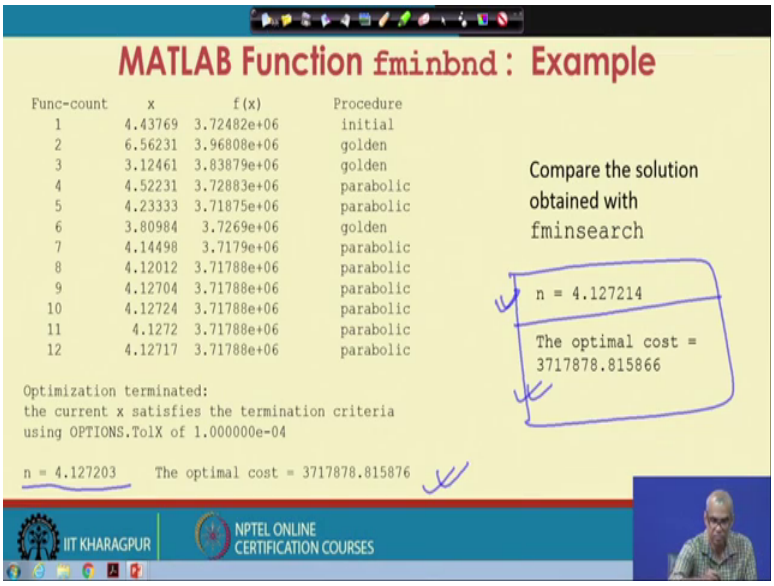
So, this is the expression of the cost. So, for given value of n, this function will be able to find the cost which is contained in C and it will be returned to the solver fminbnd. Now, we are ready to call fminbnd solver for optimization. So, again you can write these statements on MATLAB command window or you can create a script file, where you enter these statements. Save it, give a name and save it and then, run that file. So, x 1 equal to 1 and x 2 equal to 10.

So, basically we consider n is bounded between 1 and 10. Then, we use options to display the results at each iteration and also the tolerance value is set at 10 to the power minus 8 and this is the syntax we use now to call fminbnd. So, the out so, the what fminbnd will return is n which is the solution that is number of CSTR units that minimizes the total cost and f well; that means, the cost associated with this optimal number of CSTR units n.

So, arguments for fminbnd at cost; this is the function file that will evaluate the cost x 1 is lower bound set as 1 here, x 2 as upper bound set as ten here and options, then you just print the results. So, the result that you get is n equal to 4.127203 and the optimal cost is

3717878. So, you we got results very very similar to this earlier, when we use f min search. So, the optimum number of the reactors can be taken as 4 and then, you can compute the volume of each CSTR as 120 meter cube.

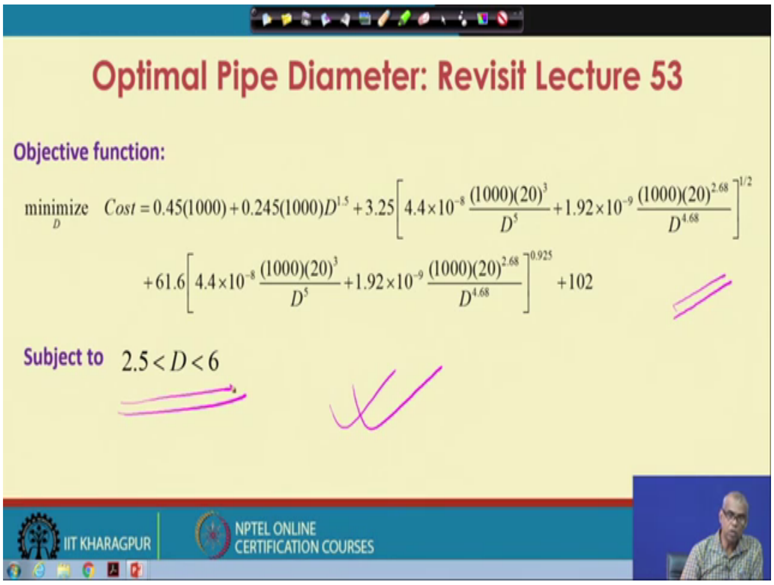
(Refer Slide Time: 22:25)



So, this is the output of fminbnd. The function count was 12 and the algorithms, again it started with golden section search; then, move to parabolic, then again it use golden section search and finally, completed with parabolic interpolation.

So, if you compare the result that was obtained with f min search, the results are here. n equal to 4.127214 was the result that we obtained with f min search and here we get 4.127203. So, up to 4 places after decimal, the results are same for value of n; same thing for the value of the optimal cost.

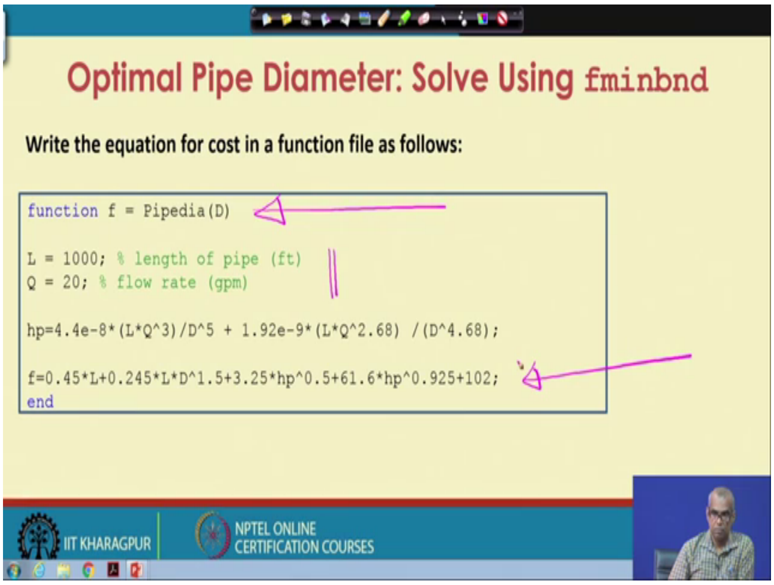
(Refer Slide Time: 24:06)



Now, as a second example, again let us re visit lecture number 53 and solve again the problem on optimal pipe diameter. Again that was formulated as a single variable optimization problem. Look at lecture 53 for the problem formulation which is reproduced here. The cost has been expressed as a single variable function of pipe diameter D by this long expression and the optimal pipe diameter has upper bound as 6 and lower bound as 2.5.

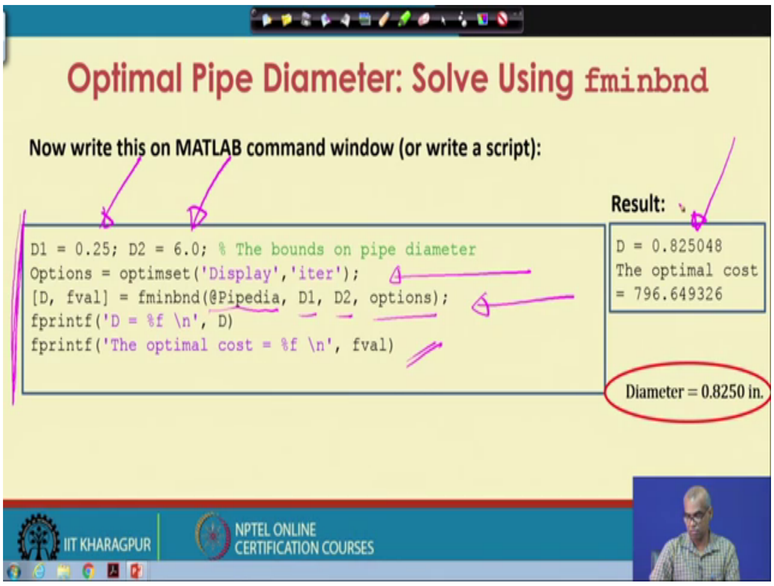
So, this can also be solved using fminbnd because it is a single variable function and we know the bounds on the optimal pipe diameter.

(Refer Slide Time: 25:30)



So, to solve using fminbnd, you first have to define the function file, where you will evaluate the objective function. So, this is true for all the functions that we have used so far. Syntax is very much similar. So, we create a function file called Pipedia which takes 1 input which is D and this stands for the diameter of the pipe. The length of the pipe and flow rate, there all specified please look at lecture number 53. And then, you can compute the cost corresponding to pipe diameter D.

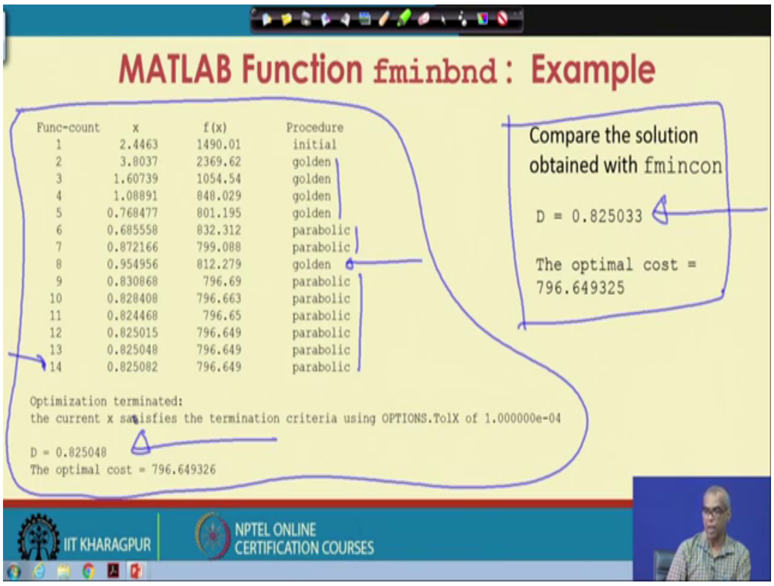
(Refer Slide Time: 26:47)



So, now once we have written the function file that will evaluate the objective function, we are ready to call the fminbnd solver. So, again you can either write these statements on the MATLAB command window or you can create a script file, write this statements, save the file you have to give a name. So, it will be your name dot in file; you have to run that file to get the solution.

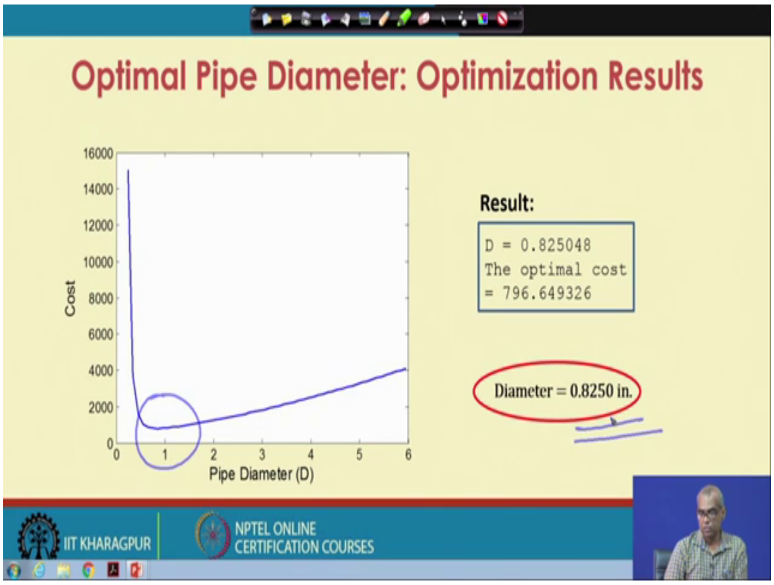
So, D 1 is 0.25 D 2 is 6, upper bound and lower bound options and then, the syntax. We are following is D and fval are the values that fminbnd will return function file at Pipedia D 1, D 2 lower bound and upper bound and options and then, you also print the optimal cost. So, the solution that you get is D equal to 0.825048 and the optimal cost associated with this pipe diameter is 796.649326.

(Refer Slide Time: 28:35)



So, this is the output of fminbnd. The function count was 14. It started with golden section search move to parabolic, again use golden section search and then finally, completed with parabolic interpolation. In our previous lecture, we solve the same problem using fmincon and again note that the values that were obtained they match up to 4 places after decimal.

(Refer Slide Time: 29:49)



Since, this was a single variable function, you can of course, plot the cost function with respect to pipe diameter and can check for yourself, the optimal solution lies around 0.8. So, the true optimal solution is computed using fminbnd is point 0.8250 inch. So, this is how we can solve a single variable minimization problem, where we know the optimal solution lies between lower bound and an upper bound. So, with this we solved, we complete lecture 53 here.